文章
Hao Ma · 一月 15, 2021 阅读大约需 5 分钟

Open Exchange

# ObjectScript错误管理

InterSystems编程语言的错误管理技术一直在发展。接下来，我们将展示几种不同的错误管理实现方式，以及为什么要使用TRY/THROW/CATCH机制。

您可以点击这里阅读官方的错误处理建议。

为了支持遗留应用程序，Inter
Systems不会废弃非推荐的错误管理方法。我们建议使用objectscriptQuality
等工具来检测遗留的非推荐用例以及其他可能的问题和错误。

## $ZERROR

$ZERROR是一种较老的错误管理机制，支持与标准“ M ”不同的实现。虽然$ZERROR现在仍然有效，但我们非常不推荐使用。

如果您已经使用了$ZERROR，那么很容易对该变量进行错误的设计使用。$ZERROR是一个全局公共变量，可以被当前进程中正在执行的所有routine（宏）（来自InterSystems或自定义的）进行访问和修改。因此，它的值仅在产生错误的时候是可靠的。InterSystems不保证$ZERROR在调用系统库时会保留旧值。

我们在这里对一些案例展开分析。

案例1：自定义代码中的错误代码

| Line | Code | Comments | $ZERROR value |
|---|---|---|---|
| 1 | Set ... | | "" |
| 2 | Set ... | | "" |
| ... | Do ... | | "" |
| ... | ... | | "" |
| **N-m** | Do CacheMethodCall() | Call to another Caché system methods | **""** |
| ... | | | "" |
| **N** | Set VarXX = MyMethod() | The custom method generates an ObjectScript error | **<UNDEFINED>B+3^Infinity Method *varPatient** |
| N+1 | Set ⋯ | | <UNDEFINED>B+3^Infinity Method *varPatient |
| **N+2** | Do OtherCacheMethodCall() | Caché system method. $ZERROR is not updated if there is no error. | <UNDEFINED>B+3^Infinity Method *varPatient |
| ... | If ... | | <UNDEFINED>B+3^Infinity Method *varPatient |
| ... | ... | | <UNDEFINED>B+3^Infinity Method *varPatient |
| ... | While ... | | <UNDEFINED>B+3^Infinity Method *varPatient |
| ... | **If $ZERROR ’ = ” ” Quit “ Error ”** | | <UNDEFINED>B+3^Infinity Method *varPatient |
| N+m | Quit "OK" | | <UNDEFINED>B+3^Infinity Method *varPatient |

案例2：内部Caché错误出现误报

在这种情况下，自定义代码运行良好，但内部Caché错误引发了一个错误。

| Line | Code | Comments | $ZERROR value |
|---|---|---|---|
| 1 | Set … | | "" |
| 2 | Set … | | "" |
| .. | Do … | | "" |
| .. | … | | "" |
| **N-m** | Do CacheMethodCall() | Internal error but it is managed internally and decides to continue execution | <UNDEFINED>occKl+3^MetodoInternoCache *o0bxVar |
| .. | | | <UNDEFINED>occKl+3^MetodoInterno *o0bxVar |
| **N** | **Set VarXX = MyMethod() // OK** | | <UNDEFINED>occKl+3^MetodoInterno *o0bxVar |
| N+1 | Set … | | <UNDEFINED>occKl+3^MetodoInterno *o0bxVar |
| N+2 | Do OtherCacheMethodCall() // OK | | <UNDEFINED>occKl+3^MetodoInterno *o0bxVar |
| … | If … | | <UNDEFINED>occKl+3^MetodoInterno *o0bxVar |
| … | … | | <UNDEFINED>occKl+3^MetodoInterno *o0bxVar |
| … | While … | | <UNDEFINED>occKl+3^MetodoInterno *o0bxVar |
| … | **If $ZERROR ’ ="" Quit "Error"** | An error is detected while there is no error at all | **<UNDEFINED>occKl+3^MetodoInterno *o0bxVar** |
| N+m | Quit "OK" | | <UNDEFINED>occKl+3^MetodoInterno *o0bxVa |

案例3：在Caché内部代码中重置$ZERROR时出现误报

在这种情况下，即使不存在错误，也会直接或间接调用一个内部Caché方法或routine（宏）来重置公共变量$ZERRO

| Line | Code | Comments | $ZERROR |
|---|---|---|---|
| 1 | Set … | | "" |
| 2 | Set … | | "" |
| … | Do … | | "" |
| … | … | | "" |
| **N-m** | **Do CacheMethodCall()** | | <UNDEFINED>occKl+3^MetodoInternoCache *o0bxVar |
| … | | | <UNDEFINED>occKl+3^MetodoInterno *o0bxVar |
| **N** | **Set VarXX = MyMethod()** | | <UNDEFINED>occKl+3^MetodoInterno *o0bxVar |
| N+1 | Set … | | <UNDEFINED>occKl+3^MetodoInterno *o0bxVar |
| N+2 | **Do OtherCacheMethodCall()** | Internal method that resets $ZERROR whether there is error or not | "" |
| … | If … | | "" |
| … | … | | "" |
| … | While … | | "" |
| … | **If $ZERROR ’ ="" Quit "Error"** | Error not detected | "" |

| Line | Code | Comments | $ZERROR |
|---|---|---|---|
| N+m | Quit "OK" | | "" |

# $ZTRAP

$ZTRAP是在上下文中进行错误管理的，因此不存在在上下文环境之外被意外覆盖的风险。当出现错误时，控件返回到调用堆栈中的第一个错误控件。

当出现错误并处理完成后，必须清除$ZTRAP，以便在发生另一个错误时避免出现无限循环。

因此，$ZTRAP在错误管理方面比$ZERROR更先进，但开发人员在添加操作时有可能会产生更多的错误。

如果想要进一步了解此方法的使用，可以查看官方文档中的 "$ZTRAP错误处理 " 节选内容。

## %Status

该方法用于系统库中，因此是调用系统库时必须使用的机制。

可以点击这里查看用法。

# TRY/THROW/CATCH

这是最现代的错误管理方法，也是目前比较推荐的方法。

您可以点击这里查看用法。

此方法可在上下文中管理错误，并且不需要开发人员管理内部错误变量。

## 优点

关于TRY/THROW/CATCH的文献有很多，这里列举了一些它的优点：

- 提供了一种清晰的方式进行异常情况处理，将错误代码与常规代码分开处理
- 简化了错误检测，所以无需在每次操作后检查错误
- 允许错误传播到上层
- 支持运行时错误，允许在崩溃后恢复并继续运行

缺点

最明显的缺点是轻微的性能损失，因此您必须知道需要在什么时候使用该方法。

通常，没有必要在每种方法上都使用TRY/THROW/CATCH，在很多情况下在操作前多进行几次简单的验证就可以避免很多避免错误，从而避免不必要地使用TRY/THROW/CATCH方法。

结论

避免使用$ZERROR和$ZTRAP。

只有在调用系统库时才使用%STATUS。

可以使用TRY/THROW/CATCH管理错误，但不要滥用。

#错误处理 #ObjectScript #Caché #InterSystems IRIS

在 InterSystems Open Exchange 上检查相关应用程序

---

**源URL:**
https://cn.community.intersystems.com/post/objectscript%E9%94%99%E8%AF%AF%E7%AE%A1%E7%90%86